ML2020SPRING HW6 Report

學號：R08946015 系級：資料科學碩一 姓名：陳鈞廷

1. 試說明 hw6_best.sh 攻擊的方法，包括使用的 proxy model、方法、參數等。此方法和 FGSM 的差異為何?如何影響你的結果?請完整討論。

這次我選擇的 hw6_best.sh 方法為 iterative FGSM，我使用的 proxy model 為 torchvision 套件中提供的 pretrained 過的 DenseNet-121，並將迭代次數設定為 2，每次迭代的 epsilon 設定為 0.05，經過 2 次的迭代 attack，pixel 的 L-∞ norm 大約為 6.0。

iterative FGSM 是使用 FGSM 迭代數次來計算出 adversarial image，每次迭代 attack 得到的 adversarial image 將作為下次迭代時 model 的 input，我認為因為 iterative FGSM 迭代了數次，所以有較大的機率可以找到（或是說組合出）較好的 noise，因此更有可能改變 top1 的預測結果。

下方表格是實驗結果，兩個方法同樣使用 DenseNet-121 作為 proxy model。可以發現在同樣的 L-∞ norm 下，iterative FGSM 有較高的成功率。

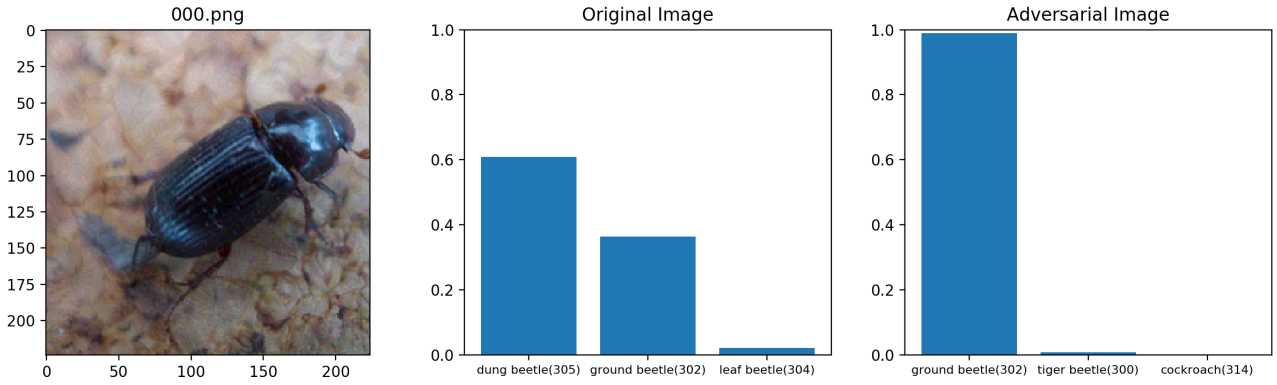
	ϵ	Success Rate	L-∞ norm
FGSM	0.1	1.00	6.00
Iterative FGSM	0.05	0.975	6.00

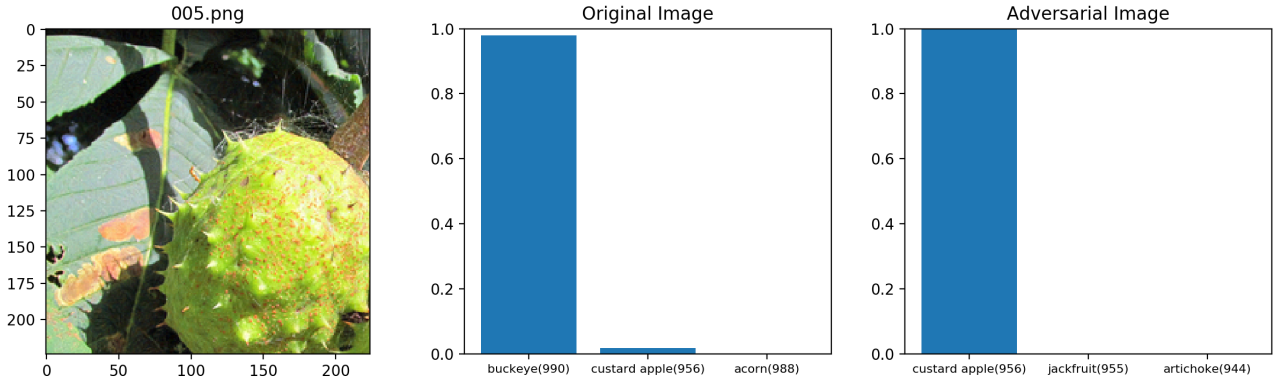
2. 請嘗試不同的 proxy model，依照你的實作的結果來看，背後的 black box 最有可能為哪一個模型?請說明你的觀察和理由。

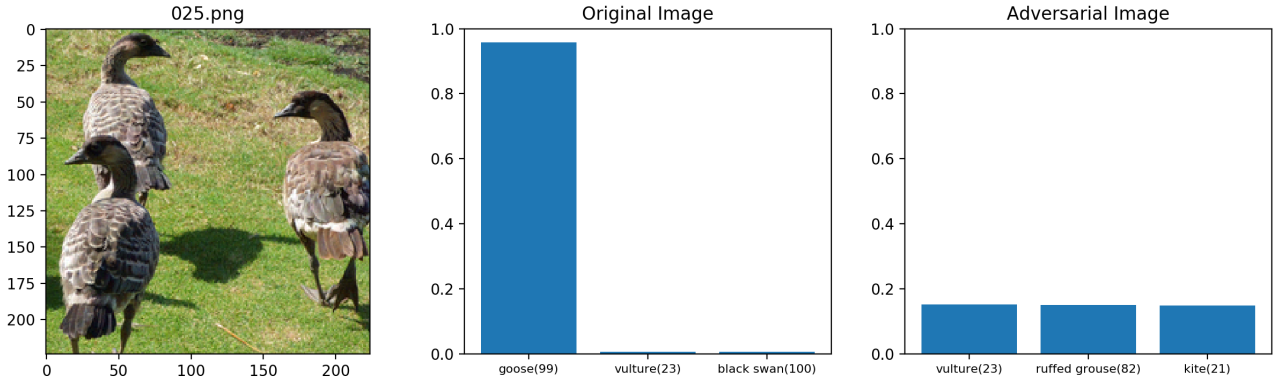
我使用了上題所描述的 hw6_best.sh 的方法針對 6 個 proxy models 來攻擊，實驗結果如下方表格所示，可以發現 DenseNet-121 的 success rate 最高，另外 DenseNet-169 的 success rate 為第二高，而我們知道相似的 model 可能會有相似的 decision boundary，也因此 adversarial transferability 會相對較顯著，因此合理推測 black box 應該是 DenseNet-121。

	VGG-16	VGG-19	ResNet-50	ResNet-101	DenseNet-121	DenseNet-169
Success Rate	0.385	0.37	0.505	0.505	1.00	0.64

3. 請以 hw6_best.sh 的方法，visualize 任意三張圖片攻擊前後的機率圖(分別取前三高的機率)。







4. 請將你產生出來的 adversarial img，以任一種 smoothing 的方式實作被動防禦(passive defense)，觀察是否有效降低模型的誤判的比例。請說明你的方法，附上你防禦前後的 success rate，並簡要說明你的觀察。另外也請討論此防禦對原始圖片會有什麼影響。

我使用了 PIL 套件中的 filter 來實現 Gaussian blurring (`PIL.ImageFilter.GaussianBlur`)，我測試了不同程度的 Gaussian blurring 套用在 adversarial image 之後的 success rate，實驗結果如下左表格所示。可以發現套用了 Gaussian blurring 後確實讓 success rate 下降，當 radius 為 2 時防禦效果最好，另外，success rate 並不會隨著 radius 上升而降

低，這可能是因為過度的模糊化會破壞圖片的重要特徵，讓辨識率下降。而下方右邊表格為 DenseNet-121 對於套用 Gaussian blurring 的原始圖片的正確辨識率，可以發現隨著模糊化程度加大，辨識率會明顯下降，因此在實作模糊瓦的防禦方法時，需要考慮適當的程度，否則會連正常圖片也失準。

	Success Rate
Adversarial Image (HW5_best)	1.000
Gaussian Blur (radius = 1)	0.960
Gaussian Blur (radius = 2)	0.695
Gaussian Blur (radius = 3)	0.715
Gaussian Blur (radius = 4)	0.815

	Accuracy
Original Images	0.925
Gaussian Blur (radius = 1)	0.840
Gaussian Blur (radius = 2)	0.595
Gaussian Blur (radius = 3)	0.350
Gaussian Blur (radius = 4)	0.210